AF/2763$

TRANSMITTAL OF APPEAL BRIEF (Large Entity)	Docket No. YO998-256

In Re Application Of: **Pednault, E.**

#11 D.Cotton 11-14-00

Serial No.	Filing Date	Examiner	Group Art Unit
09/106,784	June 29, 1998	Broda, S.	2763

Invention: **MECHANISM FOR CONSTRUCTING PREDICTIVE MODELS THAT ALLOW INPUTS TO HAVE MISSING VALUES**

OIPE JC39 NOV 0 6 2000 PATENT & TRADEMARK OFFICE

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on

RECEIVED
NOV 1 3 2000
Technology Center 2100

The fee for filing this Appeal Brief is: **$310.00**

☐ A check in the amount of the fee is enclosed.

☒ The Commissioner has already been authorized to charge fees in this application to a Deposit Account. A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **50-0510**
A duplicate copy of this sheet is enclosed.

Sean M McGinn

Signature

Dated: **November 6, 2000**

Sean M. McGinn, Esq.
Reg. No.: 34,386
Customer No.: 21254

I certify that this document and fee is being deposited on with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Signature of Person Mailing Correspondence

Typed or Printed Name of Person Mailing Correspondence

CC:

#11

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BOARD OF PATENT APPEALS AND INTERFERENCES

In re patent application of

Pednault, E.

Serial No.: 09/106,784 — Group Art Unit: 2763

Filed: June 29, 1998 — Examiner: Broda, S.

For: MECHANISM FOR CONSTRUCTING PREDICTIVE MODELS THAT ALLOW INPUTS TO HAVE MISSING VALUES

APPELLANTS' BRIEF ON APPEAL

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Appellant respectfully appeals the final rejection of claims 1-20 in the Office Action dated June 2, 2000. A Notice of Appeal was timely filed on September 5, 2000 (e.g., September 2 fell on a Saturday, September 4 was Labor Day and September 5 was the next business day).

I. REAL PARTY IN INTEREST

The real party in interest is IBM Corporation, assignee of 100% interest of the above-referenced patent application.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellant, Appellant's legal representative or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-20, all the claims pending in the application, are set forth fully in the attached Appendix.

Claims 1-20 stand rejected only under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. There are no prior art rejections. Appellant again gratefully acknowledges the Examiner's earlier indication that claims 1-2 (and presumably claims 3-20) would be allowable if the above-mentioned §101 rejection is overcome. Appellant respectfully appeals this rejection of claims 1-20.

IV. STATEMENT OF AFTER-FINAL AMENDMENT

No After Final Amendment was made. It is noted that a Response under §1.116 was filed on August 2, 2000, but no amendments were made. Therefore, the claims are pending as set forth in the Appendix.

V. SUMMARY OF THE INVENTION

The invention, as set forth and defined by independent claim 1, is a program storage device (e.g., diskette, hard drive, optical disk, etc. as is commonly known to one of ordinary skill in the art) for storing method steps of a program.

The method is for constructing predictive models that can be used to make predictions in situations where the inputs to those models can have values that are missing or are otherwise unknown.

That is, the method includes presenting a collection of training data comprising examples of input values that are available to the model together with corresponding desired output value(s) that the model is intended to predict, and generating a plurality of subordinate models (e.g., see page 19, line 26, et seq.), that together comprise an overall model. Each subordinate model has an associated set of application conditions (e.g., see page 20, lines 8-12, et seq.) that must be satisfied in order to apply the subordinate model when making predictions.

The application conditions include tests for missing values for all, some, or none of the inputs, and tests on the values of all, some, or none of the inputs that are applicable when the values of the inputs mentioned in the tests have known values.

For at least one subordinate model, the training cases (e.g., see page 20, line 14-page 23, line 9) used in the construction of that subordinate model include some cases that

indirectly satisfy the application conditions such that the application conditions are satisfied only after replacing one or more known data values in these training cases with missing values.

Further, as exemplarily defined by independent claim 1 and as described on page 23, lines 20-24), the method further includes "*outputting a specification of at least one of said subordinate models thus generated and making a prediction based on said at least one of said subordinate models thus-generated.*" In one embodiment, the specification of a plurality of subordinate models and their associated application conditions, are output to a storage device for being read by the machine, thereby <u>enabling the plurality of models to be readily applied to generate predictions</u>.

With the unique and unobvious features of the claimed invention, the method can realize significant advantages because it can be readily applied in conjunction with any known method for constructing models, including ones that require all input values to be known. Thus, the invention yields combined methods for <u>constructing models that tolerate missing values</u>.

In an exemplary embodiment, the method and storage device storing the method, can be utilized in combination with classification and regression trees, classification and regression rules, or stepwise regression (e.g., see page 9 of the present specification).

Referring to Figure 1, a Table is shown as an example of input values (training data) that are missing at random, whereas Figure 2 illustrates a Table as an example of input

values that are not missing at random. As explained in the specification (e.g., see, inter alia, pages 11-12 and page 13-14), the claimed invention yields combined methods for constructing models that tolerate missing values.

VI. ISSUES PRESENTED FOR REVIEW

The sole issue presented for review by the Board of Patent Appeals and Interferences is:

whether claims 1-20 are properly rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

VII. GROUPING OF THE CLAIMS

As supported by the following arguments, independent claim 1 and dependent claims 2-20 are each independently patentable and directed to statutory subject-matter, and do not stand or fall together.

Claim 1 recites a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for constructing a predictive model that can be used to make predictions even when the values of some or all inputs are missing or are otherwise unknown.

The claimed method includes 1) presenting a collection of training data comprising

examples of input values that are available to the model together with corresponding desired output value(s) that the model is intended to predict.

Further, the inventive method generates a plurality of subordinate models, that together comprise an overall model, in such a way that:

each subordinate model has an associated set of application conditions that must be satisfied in order to apply the subordinate model when making predictions, the application conditions comprising:

i) tests for missing values for all, some, or none of the inputs,

and

ii) tests on the values of all, some, or none of the inputs that are applicable when the values of the inputs mentioned in the tests have known values;

and

for at least one subordinate model, the training cases used in the construction of that subordinate model include some cases that indirectly satisfy the application conditions such that the application conditions are satisfied only after replacing one or more known data values in these training cases with missing values.

Lastly, the inventive method outputs a specification of at least one of the subordinate

models thus generated and making a prediction based on the at least one of the subordinate models thus-generated.

In addition, each of the dependent claims 2-20 is patently distinct from independent claim 1 from which they depend.

Each dependent claim recites additional features, not defined in independent claim 1. As discussed in greater detail below, the features defined by the dependent claims are not merely illustrations or examples, but patentable features which prevent the dependent claims from standing or falling with independent claim 1.

VIII. ARGUMENT

A. THE EXAMINER'S POSITION

As set forth on pages 2-6 of the Office Action dated June 2, 2000, the Examiner rejects claims 1-20 under 35 U.S.C. §101 under the reasoning that:

> *2.1 Product claims 1-20 are rejected because the underlying process invention comprises an abstract idea.*
>
> *2.2 Regarding Claim 1, this claim is directed to 'A program storage device readable by a machine, tangibly embodying a program of instructions*

executable by the machine to perform method steps for constructing predictive models', and the steps recited in Claim 1 describe mathematical operations comprising the abstract idea of generating models that account for missing or otherwise unknown data values.

For the purposes of examination, the 'device' of claims 1-2 will be read broadly to comprise a product claim that encompasses any and every computer implementation of a process. Neither the detailed description of the invention nor the drawings supply any tangible description of a computer implementation of the invention.

In this situation, the following paragraph in the Guidelines at IV.B.2.(a)(ii) appears controlling:

> *If a claim is found to encompass any and every product embodiment of the underlying process, and if the underlying process is statutory, the product claim should be classified as a statutory product. By the same token, <u>if the underlying Process invention is found to be non-statutory. Office personnel should classify the 'product' claim as a "non-statutory product."</u> If the product claim is classified as being a non-statutory product on the basis of the underlying process, Office personnel should emphasize that they have considered all claim limitations and are basing their finding on the analysis of the underlying process.*

[Emphasis supplied.]

Therefore, Claim 1 is rejected as being classified as a non-statutory product because the underlying process invention as claimed by Appellant is non-statutory. The method steps in Claim 1 do not: (1) recite data gathering limitations or post-mathematical operations that might independently limit the claims beyond the performance of a mathematical operation; or (2) limit the

use of the output to a practical application providing a useful, concrete, and tangible result.

2.3 Regarding Claims 2-20, the limitations supplied in these claims do not: (1) recite data gathering limitations or post-mathematical operations that might independently limit the claims beyond the performance of a mathematical operation; or (2) limit the use of the output to a practical application providing a useful, concrete, and tangible result. The analysis and conclusion regarding non-statutory subject matter is identical to Claim 1 above.

Additionally, the Examiner has analyzed some of the existing case law and on pages 3-6 in the June 2, 2000 Office Action asserts that:

Appellant's Argument

Appellant argues that the amended claims meet the requirements under 35 U.S.C. § 101 as described in AT&T Corp. v. Excel Communications Inc., 50 USPQ2d 1447 (Fed. Cir. 1999) because "Appellants [sic] submit that they have developed a useful, concrete and tangible result from the claimed features, the utility being clearly described in the application." Amendment, page 11.

Appellant further argues that the added limitation "outputting a specification of at least one of said subordinate models and making a prediction based on said at least one of said subordinate models thus generated" clearly defines "post-computation/mathematical operation processing" making amended Claim 1 statutory. See Amendment, page 12.

Examiner's Reply

The Examiner respectfully disagrees with Appellant's arguments for the following reasons.

5.1 Regarding Appellant's argument based on AT&T Corp. v. Excel Communications Inc., although the Appellant has identified a practical application (direct-mail targeted-marketing) for the method steps recited in Claim 1, the claim itself contains no corresponding limitation.

A review of the claims analyzed by the Federal Circuit in:

(1) In re Alappat, 31 USPQ2d 1545 (Fed. Cir. 1994);

(2) State Street Bank & Trust Co. v. Signature Financial Group Inc., 47 USPQ2d 1596 Fed. Cir. 1998); and

(3) AT&T Corp. v. Excel Communications Inc., 50 USPQ2d 1447 (Fed. Cir. 1999), demonstrate the differences between claims held statutory under 35 U.S.C. § 101 and claims 1-20 submitted by Appellant.

In Alappat, Claim 15 was directed at a "rasterizer for converting vector list data" and included means for "outputting illumination intensity data as a predetermined function." 31 USPQ at 1553.

The Federal Circuit held the claim as reciting "a specific machine" that produced "a concrete, and tangible result." 31 USPQ2d at 1557. Appellant's claims contain no limitation to a useful, concrete, and practical result.

In State Street, Claim 1 was directed at a "data processing system for managing a financial services configuration" and included means for processing daily asset value data and means for "allocating the percentage share that each hind holds". 47 USPQ 2d at 1599.

The Federal Circuit held that the transformation of data representing dollar amounts into a final share price, produced a "useful, concrete, and tangible result." 47 USPQ at 1601. Applicant's claims contain no similar limitation to a useful, concrete, and practical result.

In AT&T v. Excel Claim 1 was directed at a "method for use in a telecommunications system in which interexchange calls initiated by each subscriber are automatically routed over the facilities of a particular one of a plurality of interexchange carriers associated with that subscriber", and included steps of"generating a message record for an interexchange call" and including a primary interexchange indicator in each generated message record. 50 USPQ2d

at 1449.

The Federal Circuit held the claim produced a "useful, concrete, and tangible result." 50 USPQ 1452. Appellant's claims contain no similar limitation to a useful, concrete, and practical result.

5.2 Regarding Appellant's argument that amended claim 1 is statutory because the claim contains "post-computation/mathematical operation processing", this argument is rejected as the amended claim language contains no post-computer process activity but represents the output of a mathematical algorithm. As explained in State Street,

...the mere fact that a claimed invention involves inputting numbers, calculating numbers, outputting numbers, and storing numbers, in and of itself, would not render it non-statutory subject matter, unless, of course, its operation does not produce a "useful, concrete, and tangible result."

State Street at 1602. As previously explained, Appellant's claimed invention does not produce a useful, concrete, and tangible result, but describes a mathematical algorithm used to construct a predictive model. The claimed invention takes a set of data----abstract numbers----and generates mathematical models used to predict (abstract) numbers, even when some data values are unknown.

B. APPELLANTS' POSITION

1. INDEPENDENT CLAIM 1

Claim 1 recites a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for constructing <u>a predictive model that can be used to make predictions even when the values of some or all inputs are missing or are otherwise unknown</u>.

The method of claim 1 includes 1) presenting a collection of training data comprising examples of input values that are available to the model together with corresponding desired

output value(s) that the model is intended to predict.

Further, the inventive method generates a plurality of subordinate models, that together comprise an overall model, in such a way that:

each subordinate model has an associated set of application conditions that must be satisfied in order to apply the subordinate model when making predictions, the application conditions comprising:

iii) tests for missing values for all, some, or none of the inputs,

and

iv) tests on the values of all, some, or none of the inputs that are applicable when the values of the inputs mentioned in the tests have known values;

and

for at least one subordinate model, the training cases used in the construction of that subordinate model include some cases that indirectly satisfy the application conditions such that the application conditions are satisfied only after replacing one or more known data values in these training cases with missing values.

The final step of claim 1 defines outputting a specification of at least one of the subordinate models thus generated and making a prediction based on the at least one of the subordinate models thus-generated.

a. THE EXAMINER'S REJECTION IS ERRONEOUS BASED ON FACT

First, the Examiner's position is <u>flawed as a matter of fact</u>.

That is, Appellant has provided a plurality of reasons clearly establishing the statutory nature of the invention. While the Examiner presumably believes that the claims must recite the <u>exemplary application of the invention to direct-mail marketing</u> (an example listed in the specification), Appellant submits that such an amendment to the claims is unnecessary and

indeed would serve to unduly limit the invention for no apparent reason or purpose.

More specifically, the Final Office Action erroneously asserts that Appellant has disclosed methods and apparatuses for using a computer but no practical application is discussed in the specification or in the claims. Furthermore, the specification and claims merely discuss performing the abstract idea of generating models that account for missing or otherwise unknown data values. The Office Action also incorrectly asserts that no practical application of the invention is discussed, that none of the embodiments performs any post-computational processing activities, and that data is not extracted from a mathematical calculation to be manipulated to achieve a practical activity. Appellant respectfully submits that the Examiner's reasoning above and rejection are erroneous.

i. "Computer Implementations"

First, regarding the Examiner's assertion that "*[n]either the detailed description of the invention nor the drawings supply any tangible description of a computer implementation of the invention*" (Section 2.2 of the Final Office Action, second paragraph, second sentence), Appellant respectfully disagrees.

As stated on Page 19, Lines 21-24, of the detailed description of the invention, "*[t]he steps are presented in such a way that they may be readily combined with any method for constructing the subordinate models of the plurality, including ones that require all input values to be known.*"

In particular, the preferred method steps describe how to combine the invention with stepwise regression, classification and regression trees, and classification and regression rules. A deliberate effort was made to phrase the method steps in such a way that someone ordinarily skilled in the art of implementing any one of the aforementioned predictive modeling techniques could, upon reading the disclosure and the cited literature, implement the invention for that

technique.

Moreover, the method steps were phrased in a way that anticipates the possibility of combining aspects of all three types of predictive modeling techniques in a single algorithm. The purpose was to enable someone skilled in the art to apply the invention in a much broader context than is implied by any one of the afore-mentioned predictive modeling techniques.

Method Step 1 (e.g., see page 19, line 26, et seq.) is present in all three of the aforementioned predictive modeling techniques; each begins with some initial model that is then refined.

Method Steps 2a and 2b (e.g., see page 20, line 8, et seq.) address stepwise regression. This predictive modeling technique repeatedly performs incremental model refinement steps on an initial regression equation until a set of stopping conditions are met (e.g., "*until it is decided that no further refinements are justified*").

The refinement steps comprise adding variables (e.g., input data fields) to, or removing variables from, a current regression equation to produce a new regression equation. The new regression equation then becomes the current regression equation, thereby enabling further refinements to be performed. The various ways of implementing stepwise regression are well-known to those ordinarily skilled in the art of programming stepwise regression algorithms.

Stepwise regression does not consider a plurality of models, but instead repeatedly refines a single model. The detailed description of the invention teaches the advantages of maintaining a plurality of models using a regression problem as an example. Computer methods for maintaining a plurality of regression equations should be (and indeed are!) self-evident to one ordinarily skilled in the art of computer programming. For the sake of argument, maintaining and utilizing associated application conditions, on the other hand, might not be self-evident unless one is also knowledgeable about classification and regression tree algorithms and/or classification and regression rule algorithms. The cited literature on these topics teach computer methods for implementing the application conditions required by the invention. Armed with this knowledge, those skilled in the art of programming stepwise regression algorithms would then be

able to implement Method Steps 2a and 2b of the invention.

Method Step 2c (e.g., see page 21, line 16, et seq.) addresses classification and regression trees. These predictive modeling algorithms already construct pluralities of models. In this case, a plurality comprises the models at the leaves of a tree, and the application condition of each such model is the conjunction of the branch conditions along the path leading from the root of the tree to the corresponding leaf. Classification and regression tree algorithms repeatedly perform incremental model refinement steps on an initial tree (usually a single root node) until a set of stopping conditions are met (e.g., "*until it is decided that no further refinements are justified*"). Each refinement step comprises adding two or more child nodes to a leaf node in the current tree. The child nodes are assigned disjoint branch conditions and they then become new leaf nodes. The method of constructing tree branches is thus directly analogous to that described in Method Step 2c. The various ways of implementing such refinement steps are well-known to those ordinarily skilled in the art of programming classification and regression tree algorithms.

Method Step 2c specifies the preferred method of modifying classification and regression tree algorithms to incorporate the invention by specifying the preferred method for constructing tree branches using the invention.

Some classification and regression tree algorithms treat missing as a legitimate data value. Tests for missing values thus appear in various branch conditions. These algorithms exemplarily employ a version of the prior art method discussed in the Summary of the Invention beginning on Page 3, Line 15: "METHODS THAT INTRODUCE 'MISSING' AS A LEGITIMATE DATA VALUE".

To incorporate the invention in these algorithms, the same prior art methods for constructing tree branches would be used. However, as discussed in the Detailed Description of the Invention, the training cases used to construct the models that appear in each tree node would preferably be those that indirectly satisfy the application conditions of the model for those missing values that are mentioned in the application conditions and that are to be treated as missing at random.

The latter distinction is a fundamental difference between the invention and the prior art classification and regression tree methods; hence, the difference forms a basis for the patentability of the combination of the patent claims.

As discussed in the Detailed Description of the Invention, if none of the missing values mentioned in the application conditions is to be treated as missing at random, then only those training cases that directly satisfy the application conditions would preferably be used to construct the associated subordinate model, as per the prior art method. It is this use of training cases that indirectly satisfy application conditions that fundamentally distinguishes the invention from prior art methods.

Other classification and regression tree algorithms employ different methods for handling missing data. In such cases, the trees that are constructed typically do not contain tests for missing values. For such algorithms, the same prior art methods for constructing tree branches would be used, except that additional branches must be added to some tree nodes, as per the second half of Method Step 2c, in order to handle missing values using the invention.

Again, the training cases used to construct the models that appear in each tree node would preferably be those that indirectly satisfy the application conditions of the model for those missing values that are mentioned in the application conditions and that are to be treated as missing at random.

Method Step 2d (e.g., see page 22, line 28, et seq.) addresses classification and regression rules. As with classification and regression trees, these predictive modeling algorithms also construct a plurality of models. In this case, the plurality is explicitly represented as if-then rules, with the application conditions appearing in the if-parts of the rules, and the subordinate models appearing in the then-parts of the rules. When constructing rules sets, these algorithms not only consider model refinements in which the application conditions of a rule are further restricted (e.g., by adding extra application conditions as with classification and regression trees), but they also consider model refinements whose effects are to relax the application conditions of a role so that the rule is applicable in a wider range of cases (e.g., by eliminating or otherwise generalizing

one or more application conditions).

When the latter type of refinement is performed, Method Step 2d specifies that the inputs to the model that appear in the then-part of the resulting rule should preferably be restricted to those inputs that are guaranteed not to have missing values. Other than this preferred restriction, any method for relaxing application conditions can be used in conjunction with the invention.

As before, the training cases used to construct the models that appear in each rule would preferably be those that indirectly satisfy the application conditions of the rule for those missing values that are mentioned in the application conditions and that are to be treated as missing at random.

<u>Method Step 3 (e.g., see page 23, line 11, et seq.) is present in all three of the aforementioned predictive modeling techniques</u>: at some point, model refinement terminates when various stopping conditions are met.

<u>Method Step 4 (e.g., see page 23, line 14, et seq.) corresponds to the pruning operation found in classification and regression tree algorithms</u>. Computer methods for implementing post-refinement optimization (e.g., pruning) are well-known to those ordinarily skilled in the art of implementing classification and regression tree algorithms.

<u>Method Step 5 (e.g., see page 23, line 20, et seq.) can be implemented by those Ordinarily skilled in the art of computer programming</u>. Given a particular combination of data structures for representing a plurality of subordinate models, it should be self-evident how to output a specification of the plurality in such a way that the data structures can be reconstructed when the specification is inputted at a later point in time, perhaps by a separate computer program that applies the plurality to generate predictions.

As previously mentioned, the method steps were phrased in a way that anticipates the possibility of combining aspects of stepwise regression, classification and regression trees, and classification and regression rules in a single algorithm. Hence, Method Steps 2a-d cover each of the various ways of refining a model that are used in the aforementioned predictive modeling methods: adding an input to a model (Step 2a), removing an input from a model (Step 2b),

dividing the conditions under which a model is applicable into two or more subcases and building separate models for each subcase (Step 2c), and expanding the conditions under which a model is applicable (Step 2d). Which combination of Method Steps 2a-d are utilized depends on which combination of model refinements are implemented by someone ordinarily skilled in the art of constructing predictive modeling algorithms.

ii. "Useful, Concrete, Tangible Results"

Regarding the Examiner's assertion that "*[as previously explained,] Appellant's claimed invention does not produce a useful concrete, and tangible result, but describes a mathematical algorithm used to construct a predictive model*" (Section 5.2 of the Final Office Action, second to last sentence), Appellant respectfully disagrees.

The Examiner's opinion that the underlying process invention is non-statutory rests on the presumption that Step 5 of the preferred method steps does not produce useful, concrete, and tangible results:

> *Step 5 preferably comprises outputting a specification of the plurality of subordinate models' and their associated application conditions, preferably to a storage device readable by a machine, thereby enabling the plurality to be readily applied to generate predictions.* (See page 23, lines 20-24)

The Examiner's presumption contradicts common practice by those ordinarily skilled in the art of predictive modeling.

It is common practice among data analysts to separate the task of constructing predictive models from the task of applying predictive models to make predictions.

That is, after a predictive model is constructed, it is typically outputted in machine-readable form using a suitable data exchange format so that the model can then be used as input

to a computer program that applies the model to make predictions. Such outputting capability is commonly provided by predictive modeling software. Indeed, as Grossman et al. point out (Robert Grossman, Stuart Bailey, Ashok Ramu, Balinder Maihi, Michael Comelison, Philip Hallstrom, and Xiao Qin, "The Management and Mining of Multiple Predictive Models Using the Predictive Modeling Markup Language (PMML)," Armed Forces Communications and Electronics Association (AFCEA) Conference, 1999): "Ever since there has been statistical software, there has been interchange formats for predictive models."

The Predictive Modeling Markup Language (PMML) presented by Grossman et al. is only one example of an interchange format for predictive models. However, PMML is an important example in that efforts are being made to turn PMML into an open and flexible standard for exchanging predictive models among tools and applications provided by different software vendors (see http://www.dmg.org/).

The existence of predictive model interchange formats render predictive models concrete and tangible. For example, using predictive modeling software and model application software that utilize the same interchange format, one can use predictive modeling software resident on one computer to construct a predictive model and then output that model to a floppy disk. The floppy disk can then be inserted into a separate computer disconnected from the first computer, and model application software resident on the second computer can be used to apply the predictive model to data available to the second computer. The floppy disk is concrete and it tangibly embodies the predictive model.

One of the goals of the PMML standardization effort is, in fact, to enable the above scenario to be played out using predictive modeling software and model application software supplied by two independent vendors, with the PMML encoding of the predictive model preferably transferred by electronic means via a communications network instead of by physical means via a floppy disk.

Thus, the invention clearly provides a useful, concrete, and tangible result.

iii. The Examiner's Assertion directed to a "Practical Application"

The Examiner also contends that the output of the invention (e.g., a predictive model) must be limited to a practical application in order for the results (e.g., the predictive model) to be useful.

While some predictive modeling techniques may be designed for specific applications, many are not. A wide variety of predictive modeling techniques are general-purpose in nature and are utilized for specific applications by supplying the software that embodies such techniques with application-specific data. In such cases, no modifications need be made to the techniques nor the software that embodies those techniques. Moreover, the usefulness of the output model is dictated by the usefulness of the input data.

Because general-purpose predictive modeling techniques are general-purpose, they are commonly used as component technologies when building application software. This fact, in conjunction with the increasing prevalence of predictive modeling in business, has motivated Microsoft Corporation to develop their OLE DB for Data Mining (OLE DB for DM) application programming interface (API). The following excerpt from the Microsoft web document "Introduction to OLE DB for Data Mining" (http://www.microsoft.com/data/oledb/dm.htm) outlines the objectives of this API:

> *Up to now, the data mining industry has been highly fragmented, making it difficult---and costly--for application software vendors and corporate developers to integrate different knowledge-discovery tools. With the help and contributions of more than 40 ISVs in the business intelligence field, Microsoft's OLE DB for DM specification introduces a common interface for data mining that will give developers the opportunity to easily and affordably--embed highly scalable data mining capabilities into their existing applications. Microsoft's objective is to provide the industry standard for data mining so that algorithms from practically any data mining ISV can be easily plugged into a consumer application."*

An important consequence of the OLE DB for DM AP1 is that it effectively commoditizes predictive modeling software by separating such software from the applications that use it. The API thereby enables predictive modeling software provided by one vendor to be substituted for predictive modeling software provided by another vendor without significant changes to the underlying application software.

The commoditization of general-purpose predictive modeling technology implies that the usefulness of such technology is not tied to any specific application. As stated throughout the patent application, the invention is widely applicable and has great general utility. In particular, it can be combined with general-purpose predictive modeling techniques such as stepwise regression, classification and regression trees, and classification and regression roles (e.g., see page 9, lines 11-15). Hence, the usefulness of the invention is likewise not tied to any specific application and certainly not to direct-mail marketing. To require such language in the claim would be analogous to requiring a patent to an automobile to include claim language limiting the automobile to driving on a particular street! Obviously, such requirement is unreasonable to the point of making any subsequently-issued patent worthless!

Thus, in the Final Office Action, Appellant submits that the Examiner is erroneous in his reasoning. That is, as mentioned above, the Examiner asserts:

> *Regarding Applicant's argument based on AT&T Corp. v. Excel Communications Inc., although the Applicant has filed a practical application (direct-mail targeted-marketing) for the method steps recited in Claim 1, the claim itself contains no corresponding limitation.*
>
> *A review of the claims analyzed by the Federal Circuit in:*
>
> *(1) In re Alappat, 31 USPQ2d 1545 (Fed. Cir. 1994);*
>
> *(2) State Street Bank & Trust Co. v. Signature Financial Group, 47 USPQ2d 1596 (Fed. Cir. 1998); and*

(3) AT&T Corp. v. Excel Communications Inc., 50 USPQ2d 1447 (Fed. Cir. 1999), demonstrate the differences between claims held statutory under 35 U.S.C. Section 101 and claims 1-20 submitted by Applicant.

The Federal Circuit held the claim produced a "useful, concrete, and tangible result." 50 USPQ 1452. Applicant's claims contain <u>no similar limitation</u> to a useful, concrete, and practical result." (Emphasis Appellant's).

Upon careful examination of the Federal Circuit's decisions and its reasoning in the above cases, Appellant finds no requirement expressed or implied by the Federal Circuit that a claim must be limited to a specific application --- such as direct-mail targeted-marketing --- in order for that claim to be held statutory under 35 U.S.C. § 101.

The only requirement expressed by the Court is that the claimed invention must produce a useful, concrete, and tangible result.

Appellant has argued that the claimed invention, taken as a whole, does in fact produce a useful, concrete, and tangible result: namely, a <u>predictive model that can be used to make predictions even when the values of some or all of its inputs are missing or are otherwise unknown</u>.

In response to Appellant's argument, the Examiner has simply asserted that "Applicant's claimed invention does not produce a useful, concrete, tangible result, but describes a mathematical algorithm used to construct a predictive model."

The Examiner has offered no explanation as to why he considers a predictive model is not a useful, concrete, and tangible result.

Instead, the Examiner simply contends that the process is nothing more than a mathematical algorithm.

With regard to mathematical algorithms, the Federal Circuit has clearly established the following standard for identifying unpatentable mathematical algorithms:

"In Diehr, the Court explained that certain types of mathematical subject matter, standing alone, represent nothing more than abstract ideas until reduced to some type of practical application, i.e., "a useful, concrete and tangible result." Alappat, 33 F.3d at 1544, 31 USPQ2d at 1557.

"Unpatentable mathematical algorithms are identifiable by showing that they are merely abstract ideas constituting disembodied concepts or truths that are not "useful." From a practical standpoint, this means that to be patentable an algorithm must be applied in a "useful" way." State Street v. Signature Bank, 47 USPQ2d at 1600, 1601.

Appellant submits that the claimed invention is not an "abstract idea constituting disembodied concepts or truths that are not "useful."

Rather, the invention constitutes a practical application of mathematical principles to achieve a useful, concrete, and tangible result.

Moreover, the usefulness of the invention is not restricted to specific applications, such as targeted marketing. Predictive modeling technology, in general, and Appellant's invention, in particular, are useful in a very wide range of applications. For example, the UCI Machine Learning Repository (accessible over the Internet at http://www.ics.uci.edu/~mlearn/MLSummary.html) contains over 100 databases that are used by the academic community to evaluate machine learning and predictive modeling algorithms. Each database represents a different specific application.

To apply machine learning and predictive modeling algorithms in specific applications, one simply supplies the algorithms with application-specific data. The step of applying the resulting models to generate predictions for intended applications is conventional, obvious, and noninventive to those skilled in the art of predictive modeling.

Appellant submits that predictive modeling technology, in general, and Appellant's invention, in particular, should be viewed in the same light as spreadsheet programs and relational database management systems. The usefulness of these latter two inventions

transcends specific applications, and so too does predictive modeling technology, in general, and Appellant's invention, in particular.

The first spreadsheet program (i.e., VisiCalc) literally brought early personal computers (i.e., the Apple IIe) out of the basements of hobbyists and into the offices of corporations. Spreadsheet programs had this effect precisely because they could be used for many different business purposes --- they were not application-specific.

The wide-ranging usefulness of spreadsheet programs created such a large market demand that the demand literally launched the personal computer revolution.

Relational database management systems are likewise not tied to specific applications. Relational databases provide all the necessary functionality for storing, retrieving, and querying large repositories of data without placing any restrictions on the nature of the data or on the database transactions that are to be performed. Like spreadsheet programs, the usefulness of relational database management systems is enhanced many fold by the very fact that they have multiple uses. The usefulness of relational databases as perceived by the marketplace is evidenced by the fact that Larry Ellison, the head of Oracle Corporation, a leading provider of relational database management software, is now the second wealthiest person in the U.S. as a result of his Oracle holdings with a net worth of $58 billion (just $5 billion behind Bill Gates).

Spreadsheet programs and relational database management systems are undeniably useful in the sense intended by 35 U.S.C. § 101.

Their usefulness derives from the fact that the processes that each employs are not specific to particular applications, but are generic to a wide range of applications. However, the fact that these processes are generic in nature does not imply that the processes are "merely abstract ideas constituting disembodied concepts or truths that are not "useful." The generic nature of the processes only implies that the processes can be used in a wide range of applications.

Appellant submits that predictive modeling technology, in general, and his invention, in particular, likewise employ processes that are generic in nature. Appellant also submits that, as

with spreadsheet programs and database systems, the fact that Appellant's processes are generic in nature likewise does not imply that these processes are "merely abstract ideas constituting disembodied concepts or truths that are not "useful," it only implies that the processes can be used in a wide range of applications.

Predictive modeling technology is increasingly being used to address the problem of extracting useful information from large volumes of data now being collected and stored in database systems:

"The corporate, governmental, and scientific communities are being overwhelmed with an influx of data that is routinely stored in on-line databases. Analyzing this data and extracting meaningful patterns in a timely fashion is intractable without computer assistance and powerful analytical tools. Standard computer-based statistical and analytical packages alone, however, are of limited benefit without the guidance of trained statisticians to apply them correctly and domain experts to filter and interpret the results. The grand challenge of knowledge discovery in databases is to automatically process large quantities of raw data, identify the most significant and meaningful patterns, and present these as knowledge appropriate for achieving the user's goals." C.J. Matheus, P.K. Chan, and G. Piatetsky-Shapiro, "Systems for Knowledge Discovery in Databases," IEEE Transactions on Knowledge and Data Engineering, Special Issue on Learning and Discovery in Knowledge-Based Databases, Vol. 5, No. 6, pp. 903, December 1993.

Predictive modeling technology is specifically directed toward automatically extracting meaningful patterns from data: specifically, patterns that have predictive value.

Because the knowledge discovery problem is broad in scope, any technology developed to address this problem should ideally be generic in nature, and not specific to particular Applications.

<u>In other words, creating widely applicable, application-independent technology is an explicit design consideration for enhancing the usefulness of the technology.</u>

The prior art cited in Appellant's specification is itself generic in nature, as are the improvements to that prior art that constitute Appellant's invention. The usefulness and broad

applicability of the prior art can be demonstrated by way of example, described below with regard to constructing predictive models for housing data from a particular location. The example also serves to demonstrate the utility of the invention in such an application. Obviously, as the Examiner will recognize, applying the invention to housing data (as but one exemplary application of many) is a far cry from direct mail marketing.

Tables 1-3 (see attached Exhibits 1-3) show a sample from a data set commonly known within the predictive modeling community as the "Boston Housing Data" (D. Harrison and D.L. Rubinfield, "Hedonic prices and the demand for clean air," Journal of Environmental Economics and Management, Vol. 5, pp 81-102, 1978). This is one of the data sets available from the UCI Machine Learning Repository previously cited. Harrison and Rubinfield collected and analyzed this data to determine whether air pollution had any effect on house values within the greater Boston area.

Figure 1 (see attached Exhibit 4) shows a decision tree generated using the CART algorithm (L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and regression trees, New York: Chapman & Hall, 1984) as implemented in STATISTICA for Windows (STATISTICA for Windows [Computer program manual], Version 5.5, 1995, StatSoft, Inc., 2300 East 14th Street, Tulsa, OK, 74104-4442, http://www.statsoft.com).

The program was told to construct a decision tree model that predicts PRICE (i.e., the median value of owner-occupied homes broken down into high, medium, and low ranges) using all of the other columns in the data table as potential inputs to the model. Each node in the tree corresponds to a subset of the data and is represented diagrammatically as a numbered box. Each node also contains a histogram of the proportion of high-, medium-, and low-priced neighborhoods that belong to the corresponding subset of data, and each is also labeled with the dominant price range within that subset.

Tree branches correspond to tests on the values of the inputs to the model and it is these tests that define the subsets of data that correspond to each node in the tree. Left-going branches are followed when the outcome of a test is "yes" or "true;" right-going branches are followed

when the outcome of a test is "no" or "false." Node 1 is the root of the tree and it corresponds to the entire set of data. Node 2 corresponds to the subset of data for which %LOWINCM is less than or equal to 14.4. Node 5 corresponds to the subset of data for which %LOWINCM is less than or equal to 14.4 and AVGNUMRM is greater than 6.527, and so on. The leaves of the tree correspond to the predictions made by the decision tree model.

Figure 1 (see attached Exhibit 4) demonstrates the ability of decision tree algorithms to automatically extract meaningful patterns from a collection of data.

As the tree model indicates, air pollution does have an effect on house prices, but only for neighborhoods having a sufficiently large percentage of low-income housing. For all other neighborhoods, house prices are primarily affected by the size of the house, as indicated by the average number of rooms per house in the neighborhood. When air pollution is a factor, but the air pollution level is sufficiently small, then the next most important factor affecting house prices is the racial skew of the neighborhood, which is measured as the squared difference between the racial mix of the neighborhood versus the average racial mix of the greater Boston area population as a whole.

Aside from possibly being "politically incorrect," this measurement could very well be masking a more complicated underlying reason for price differences.

To demonstrate how decision tree algorithms can be used as an investigative tool to dig deep into data to uncover more complicated relationships, the program was executed again, but this time it was told to predict PRICE using all of the other data columns except RACESKEW as potential inputs. Figure 2 (see attached Exhibit 5) shows the resulting decision tree. As this tree model indicates, when air pollution is a factor, but the air pollution level is sufficiently small, then the next most important factors affecting house prices are crime rate, the percentage of non-retail industrial land, and the distance to a major center of employment, with the more desirable (i.e., higher-priced) neighborhoods being those with low crime rates (i.e., node 8) and those with sufficiently large percentages of non-retail industrial land located away from centers of employment (i.e., node 13).

To demonstrate that decision tree algorithms are not application-specific, but can be applied to any application simply by providing application-specific data as input, the program was executed again, but this time it was told to predict the air pollution level (NOXLEVEL) using all of the other data columns as potential inputs, including PRICE. Figure 3 (see attached Exhibit 6) shows the resulting tree. As this tree illustrates, the majority of neighborhoods having the highest levels of air pollution (i.e., node 13) are those with sufficiently large percentages of non-retail industrial land, sufficiently large percentages of older buildings, and sufficiently high tax rates. Not surprisingly, these factors characterize downtown Boston and its immediate vicinity. The majority of neighborhoods having the lowest levels of air pollution (i.e., node 11) are those that have sufficiently small percentages of non-retail industrial land, sufficiently large percentages of houses on large lots, and are sufficiently far from centers of employment. These characteristics are typical of outlying suburbs. The majority of neighborhoods having moderate levels of air pollution (i.e., node 14) are those with sufficiently small percentages of non-retail industrial land, sufficiently small percentages of houses on large lots, and easy access to radial highways that lead into Boston. These characteristics are typical of urban residential neighborhoods favored by commuters.

Although the relationships described above make intuitive sense once the trees in Figures 1-3 (Exhibits 4-6) are examined in detail, it is important to keep in mind that the program itself has no knowledge of these intuitions nor of the source of data. The program is merely analyzing the data to identify patterns that have predictive value.

Nevertheless, the program produces meaningful results.

The usefulness of decision tree algorithms, and automated predictive modeling algorithms in general, derives from the fact that they can perform their analyses automatically without human intervention, and without being told what kinds of relationships to look for. All that they need to be told is which data values are to be predicted, and which data values can be used as inputs to make those predictions.

The generic nature of decision tree algorithms makes them extremely useful for the

purpose of knowledge discovery in databases.

The examples presented above clearly establish that decision tree algorithms are not "merely abstract ideas constituting disembodied concepts or truths that are not "useful."" The examples demonstrate that the decision tree models that are produced as output are useful, concrete, and tangible results that have specific meaning with respect to the input data and the modeling objectives (i.e., which data element to predict in terms of which other data elements). Hence, the generic nature of decision tree algorithms does not imply that they are "mathematical algorithms" in the sense defined by the Federal Circuit, it only implies that these algorithms can be used in a wide range of applications.

Appellant's contention that decision tree algorithms are not "mathematical algorithms" in the sense defined by the Federal Circuit is further corroborated by the fact that numerous U.S. Patents for decision tree algorithms have been issued in which the claims contain no limitations to specific applications. For example, the following patents have issued:

6,058,205 issued 05/02/2000: "System and method for partitioning the feature space of a classifier in a pattern classification system";

6,055,539 issued 04/25/2000: "Method to reduce I/O for hierarchical data partitioning methods";

6,026,399 issued 02/15/2000: "System and method for selection of important attributes";

5,982,934 issued 11/09/1999: "System and method for distinguishing objects";

5,899,992 issued 05/04/1999: "Scalable set oriented classifier";

5,870,735 issued 02/09/1999: "Method and system for generating a decision-tree classifier in parallel in a multi-processor system";

5,799,311 issued 08/25/1998: "Method and system for generating a decision-tree classifier independent of system memory size";

5,787,274 issued 07/28/1998: "Data mining method and system for generating a decision tree classifier for data records based on a minimum description length (MDL) and presorting of records";

5,694,524 issued 12/02/1997: "System and method for identifying conditions leading to a particular result in a multi-variant system";

4,719,571 issued 01/12/1988: "Algorithm for constructing tree structured classifiers"

U.S. Patents have also been issued for decision rule algorithms in which the claims likewise contain no limitations to specific applications (decision rules generalize decision trees by allowing logical overlaps among rules, whereas in decision trees each leaf corresponds to a rule and these rules are mutually exclusive);

5,802,509 issued 09/01/1998: "Rule generation system and method of generating rule";

5,761,389 issued 06/02/1998: "Data analyzing method and system";

5,740,323 issued 04/14/1998: "Evolutionary adaptation type inference knowledge extracting apparatus capable of being adapted to a change of input/output date and point of sales data analyzing apparatus using the apparatus";

5,727,199 issued 03/10/1998: "Database mining using multi-predicate classifiers";

5,719,692 issued 02/17/1998: "Rule induction on large noisy data sets";

U.S. patents have also been issued for predictive modeling algorithms that are closely related to decision tree and decision rule algorithms in which the claims similarly contain no limitations to specific applications;

6,009,239 issued 12/28/1999: "Inference apparatus and method for processing instances using linear functions of each class";

5,809,499 issued 09/15/1998: "Computational method for discovering patterns in data sets";

5,627,945 issued 05/06/1997: "Biased learning system";

5,481,650 issued 01/02/1996: "Biased learning system";

Appellant's invention addresses the problem of how to handle missing data values when constructing and applying predictive models. Numerous U.S. Patent have been issued for algorithms related to this problem in which the claims similarly contain no limitations to specific applications. For example, the following patents have issued:

6,047,287 issued 04/04/2000: "Iterated K-nearest neighbor method and article of manufacture for filling in missing values";

5,835,902 issued 11/10/1998: "Concurrent learning and performance information processing system";

5,842,189 issued 11/24/1998: "Method for operating a neural network with missing and/or incomplete data";

5,819,006 issued 10/06/1998: "Method for operating a neural network with missing and/or incomplete data";

5,802,256 issued 09/01/1998: "Generating improved belief networks";

5,748,848 issued 05/05/1998: "Learning method for a neural network";

5,729,661 issued 03/17/1998: "Method and apparatus for preprocessing input data to a neural network";

5,706,401 issued 01/06/1998: "Method for editing an input quantity for a neural network";

5,704,018 issued 12/30/1997: "Generating improved belief networks";

5,704,017 issued 12/30/1997: "Collaborative filtering utilizing a belief network";

5,696,884 issued 12/09/1997: "Method for assisting in rendering a decision using improved belief networks";

5,613,041 issued 03/18/1997: "Method and apparatus for operating neural network with missing and/or incomplete data";

5,448,684 issued 09/05/1995: "Neural network, neuron, and method for recognizing a missing input value";

The invention filed by Appellant is an improvement to the processes employed by decision tree algorithms, decision rule algorithms, and stepwise regression algorithms.

In as much as these processes are not "mathematical algorithms" in the sense defined by the Federal Circuit, Appellant submits that his improvement to these processes is likewise not a "mathematical algorithm."

Appellant's invention exploits the fact that these three classes of algorithms employ the same basic process of starting with an initial model and then incrementally refining that model until a suitable stopping criterion is met.

An overview of stepwise regression can be found in the on-line statistics textbook provided over the Internet as a public service by StatSoft, Inc. (http://www.statsoft.com/textbook/stathome.html):

"Stepwise model-building techniques for regression designs with a single dependent variable are described in numerous sources (e.g., see Darlington, 1990; Hocking, 1966, Lindeman, Merenda, and Gold, 1980; Morrison, 1967; Neter, Wasserman, and Kutner, 1985; Pedhazur, 1973; Stevens, 1986; Younger, 1985). The basic procedures involve (1) identifying an initial model, (2) iteratively "stepping," that is, repeatedly altering the model at the previous step by adding or removing a predictor variable in accordance with the "stepping criteria," and (3) terminating the search when stepping is no longer possible given the stepping criteria, or when a specified maximum number of steps has been reached." (http://www.statsoft.com/textbook/stgsr.html#stepwise).

Additional details on the individual method steps of stepwise regression likewise appear in the cited article.

An overview of decision tree algorithms and various prior art methods of handling missing values can found in the paper by J.R. Quinlan cited in the patent application:

"The 'standard' technique for constructing a decision tree classifier from a training set of cases with known classes, each described in terms of fixed attributes, can be summarized as follows:

* If all training cases belong to a single class, the tree is a leaf labeled with that class
* Otherwise,
  - select a test, based on one attribute, with mutually exclusive outcomes;
  - divide the training set into subsets, each corresponding to one outcome; and
  - apply the same procedure to each subset. "(J.R. Quinlan, "Unknown attribute values in

induction," Proceedings of the Sixth International Machine Learning Workshop, pp 164, Morgan Kaufmann Publishers, 1989).

Details on the individual method steps performed by decision tree algorithms can be found in another article in StatSoft's on-line textbook (http://www.statsoft.com/textbook/stclatre.html#computation).

Decision rule algorithms tend to be more varied in their design, but they likewise perform iterative refinement operations. One of the important incremental refinement operations found in most decision rule algorithms is to relax the application conditions of a rule so that the rule is applicable in a wider range of cases (e.g., by eliminating or otherwise generalizing one or more application conditions). Detailed overviews of decision rule algorithms can be found the paper by P. Domingos cited in Appellant's specification (P. Domingos, "Unifying instance-based and rule-based induction,"Machine Learning, Vol. 24, pp 141-168, 1996), and in U.S. Patent No. 5,719,692, "Rule induction on large noisy data sets."

As previously stated, Appellant's invention exploits the fact that these three classes of algorithms --- decision trees, decision rules, and stepwise regression --- employ the same basic process of starting with an initial model and then incrementally refining that model until a suitable stopping criterion is met.

Appellant's detailed specification begins with a description of the underlying principle that the invention embodies. Although this principle is itself abstract, Appellant's invention, by contrast, is a concrete application of the principle to achieve a useful end: namely the construction of predictive models that are capable of generating reliable predictions even when the values of some model inputs are missing or are otherwise unknown.

Appellant's invention is first introduced by way of two simple examples. The examples are contrived to provide a clear illustration to those skilled in the art of predictive modeling of how the underlying principle of the invention is applied using Appellant's method. The examples are very simple so as not to confuse the reader with distracting details that are not pertinent to the teaching of the invention.

The first example illustrates one of the important aspects of Appellant's invention: namely, the step of training a model using training cases that either directly or indirectly satisfy the application conditions of that model. This step is inventive and distinguishes Appellant's invention from prior art methods.

The purpose of this inventive and distinguishing step is to obtain more accurate estimates of model parameters. In the case of the example, the model parameter in question is the mean of Y. For other types of models, other model parameters would be involved. However, the purpose of this step and the result it achieves remains the same regardless of the type of model being considered: which is to obtain more accurate estimates of the model parameters --- and, hence, a more accurate model.

The second example illustrates that models should be trained using training cases that indirectly satisfy the application conditions of the model only in those situation in which missing values are noninformative; that is, when values are missing for random reasons.

Thus, the inventive and distinguishing step in Appellant's invention must be applied conditionally.

Appellant then presents method steps for a computer-implementable process for determining which missing values are informative and which are missing at random. This process can optionally be used in combination with Appellant's method for constructing predictive models. It involves applying the latter method many times over using different assumptions with regard to which missing values are informative and which are missing at random, and then choosing the combination of assumptions that yield the best overall predictive model.

Appellant then presents the preferred method steps for constructing predictive models using his invention. As stated in the specification, "[t]he steps are presented in such a way that they may be readily combined with any method for constructing the subordinate models of the plurality, including ones that require all input values to be known."

In particular, because the preferred method steps have the same overall structure as do

stepwise regression algorithms, classification and regression tree algorithms, and classification and regression rule algorithms, the method steps can be combined with any of these algorithms.

For example, with regard to the Boston Housing example presented earlier, if the preferred method steps were combined with the Statistica algorithm used to generate the decision tree in Figure 3 (Exhibit 6), then Method Step 2(c) would call for additional branches to be added to Nodes 1, 2, 3, 4, 5, and 7 to cover the cases in which %INDUSTY, %BIGLOTS, %OLDBLDG, HWYACCES, DIST2WRK, and TAX_RATE, respectively, have missing values. On the other hand, because Node 9 can be reached only when the value of %INDUSTY is not missing, Step 2(c) specifies that no additional branches would have to be added to that node. Note that, in this case, the subordinate models are the models in the leaves of the evolving decision tree. The application conditions for each subordinate model are the conjunction of the branch conditions leading from the root of the tree to the corresponding leaf.

The preferred method steps thus describe concrete improvements to the prior art when practiced in combination with the prior art.

In presenting the preferred method steps, the inventive step of training subordinate models using training cases that either directly or indirectly satisfy the application conditions of that model is specified in the preamble of the preferred method steps because this inventive step is applied conditionally depending on the application conditions of each subordinate model, and on whether the corresponding missing values (if any) are to be treated as missing at random.

If a process for constructing predictive models calls for this inventive step to be applied, then that process is applying Appellant's art. On the other hand, if this step is not applied, then the process is not applying Appellant's art.

Claim 1 recites Appellant's inventive and distinguishing step within the context in which in which it makes sense to apply that step: that is, as part of the step of generating a plurality of subordinate models and associated application conditions.

Appellant submits that Claim 1 clearly defines the metes and bounds of Appellant's

claimed invention, and that it does not claim beyond that which Appellant has invented.

Inasmuch as Appellant's specification and claims also pass the hurdles of usefulness, novelty, nonobviousness, and enablement, Appellant submits that Claim 1 and its dependent claims are deserving of patent protection.

Moreover, the non-obvious and unique combination of features provides a method stored on the claimed program product which solves prediction problems in many applications involving data with missing values (again, see the present Application, page 9, lines 4-15), and thus has great utility and is concrete.

Thus, the invention clearly is a statutory product and embraces statutory subject matter clearly worthy of a U.S. Letters Patent.

In view of the foregoing, reconsideration and withdrawal of the rejection is respectfully requested.

B. THE REJECTION IS ERRONEOUS AS A MATTER OF LAW

Secondly, as is believed clear in all the preceding discussion, <u>the Examiner's position is flawed as a matter of law</u>.

That is, Appellant submits that the Office Action references an improper standard with respect to 35 U.S.C. §101.

Appellant again notes the Federal Circuit's decision in <u>AT&T Corp v. Excel Communications</u>, 50 USPQ2d 1447 (Fed. Cir. 1999) (hereafter AT&T v. Excel). This case discusses the current status of 35 U.S.C. §101. <u>Id</u> at 1451. The Federal Circuit states that a process that applies an equation to a new and useful end is at the very least not barred by the threshold by §101. Furthermore, a claimed processing system for implementing a financial management system (as in <u>State Street</u>) constituted a practical application of a mathematical algorithm by producing "a useful, concrete and tangible result." <u>Id</u> at 1451. Furthermore, in discussing <u>State Street</u>, the Federal Circuit held that there was patentable subject matter because the system takes data representing discrete dollar amounts through a series of mathematical

calculations to determine a final share price, which was considered a useful, concrete and tangible result. See page 1452.

The Court also suggests that the notion of a physical transformation (as alluded to in the Final Office Action) is but one example of how a mathematical algorithm may bring about a useful application. Therefore, Appellant submits that the Office Action makes an improper rejection under 35 U.S.C. §101. Patentability under 35 U.S.C. §101 requires a determination of whether a useful, concrete and tangible result is accomplished by the claimed features. As discussed above, Appellant submits that it has developed a useful, concrete and tangible result from the claimed features, the utility being clearly described in the application as discussed above.

As mentioned above, the present invention relates to method for constructing a predictive model that can be used for making predictions (e.g., reliable ones on which a decision may be based) even when the values of some or all inputs are missing or are otherwise unknown. See pages 1, lines 9-12; page 2, lines 9-14, etc.). Accordingly, the present invention provides a predictive model which is superior to the prior art and which can make superior predictions from those of the prior art models, even when some or all data values are unknown or missing. This obviously relates to making real-world predictions for real-world problems and decisions. Indeed, on pages 1-2 a real world, exemplary application of direct-mail targeted-marketing purposes in industries that sell directly to consumers.

Clearly, such an application is a useful, concrete and tangible result especially in the direct-mail industry. By the same token, contrary to the Examiner's belief, Appellant submits that the claims need not specifically recite such an exemplary application, thereby limiting the claims only to such an exemplary application. Indeed, to do so would be improper (and foolhardy) for Appellant. As mentioned above in the analogy to an automobile patent, such an erroneous requirement by the Examiner would render any patent relatively useless.

Indeed, the PTO (Manual of Patent Examining Procedure, Seventh Edition, Revision 1, February 2000), states:

> *"A claim is limited to a practical application when the method, as claimed, produces a concrete, tangible and useful result; i.e., the method recites a step or act of producing something that is concrete, tangible and useful. See AT&T 172 F.3d at 1358, 50 USPQ2d at 1452." Manual of Patent Examining Procedure, Section 2106, page 2100-15, Rev. 1, Feb 2000.*

While Appellant recognizes that this statement is consistent with the current state of the law, neither this statement nor the case law implies that the method steps must LIMIT THE USE of any concrete, tangible, and useful results that are produced as output by the claimed method, as the Examiner contends.

Thus, the Examiner is misinterpreting both the law and current U.S.P.T.O. procedure as set forth in the M.P.E.P.

Additionally, as described above, the predictive models produced by the method of the invention are concrete, tangible and useful results, and hence the claimed invention produces concrete, tangible and useful results.

Furthermore, the present application discusses the problems of the prior art and how the present invention overcomes such problems (and in some cases can be used with the conventional methods of model generation) (e.g., see pages 2-7 and 23-26 of the present application).

Pages 10-22 describe features of the present invention and the respective features that obtain the objectives of the present application. The detailed description discusses the use of the invention in the form of a special apparatus or computer program executed in a generally used computer (e.g., see page 19, lines 22-24). One skilled in the art would clearly understand that this stored data may be read from the computer such as on a display unit, an output unit such as a printer, etc. Clearly, the generation of such predictive models provides a useful, concrete and tangible result for at least the direct-marketing industry (e.g., see page 1, line 29 to page 2, line 4). Indeed, such predicative models allow predictions to be generated with increased reliability despite their being missing values (e.g., possibly missing demographic, credit or other data

inputs), and allows a greater return on marketing investments in this particular application.

The claims describe the program storage device for storing the method for constructing predictive models that can be used for making such predictions despite the presence of missing data values. The generation of a predictive model is patentable subject matter if it is a practical application that produces a useful, concrete and tangible result. See AT&T v. Excel.

It is clear that this invention as a whole is applied in a useful manner, as described above (i.e., it is useful to generate a predictive model which will generate predictions having increased reliability and upon which marketing and financial decisions may be made). The independent claims set forth very detailed steps of how to arrive at this result so as to avoid problems of the prior art.

There is no requirement that the claims set forth any post-computational activity as asserted in the Office Action. Rather, as discussed on page 1452 of AT&T v. Excel, a physical transformation is merely one example of how a mathematical algorithm may bring about a useful application. In this application, the construction of the subordinate models (e.g., as defined in (2) in independent claim 1) and specifically the testing of inputs, and treatment of known data values and missing values in the claim results in the construction of a model which has increasingly reliable predictions as compared to the prior art and which avoids the problems of the prior art. By avoiding these problems, the present application provides a useful, concrete and tangible result and therefore the requirements of 35 U.S.C. §101 are met.

Additionally, Appellant again points out that claim 1 recites, inter alia, "*outputting a specification of at least one of said subordinate models thus generated and making a prediction based on said at least one of said subordinate models thus-generated*". This clearly defines a "post-computation/mathematical operation processing" and clearly the subject matter of independent claim 1 (and claims 2-20 which depend from claim 1) is statutory and allowable over the prior art of record.

Therefore, in view of all of the foregoing, the claimed invention of independent claim 1 is indeed directed to statutory subject matter within the meaning of 35 U.S.C. §101.

2. DEPENDENT CLAIMS

While independent claim 1 is directed to statutory subject matter, as discussed above, similarly dependent claims 2-20 define similar statutory subject matter separately and distinctly from independent claim 1 as these dependent claims recite additional elements clearly providing useful, concrete and tangible results.

For example, claim 2 recites "wherein step (2) comprises generating a plurality of subordinate models such that the plurality cannot be arranged into a decision-tree hierarchy in such a way that:

(1) each branch of the tree corresponds to a test on the values of one or more data fields that can be satisfied only when those data fields have known values;

(2) each leaf of the tree corresponds to a subordinate model whose application conditions are defined by the conjunction of the tests along the branches that lead from the root node of the tree to the leaf node;

(3) the root node of the tree corresponds to a subordinate model whose application conditions include missing-value tests for the data fields mentioned in the tests associated with the tree branches that emanate from the root node;
and

(4) each interior node of the tree other than the root node corresponds to a subordinate model whose application conditions are defined by the conjunction of the tests along the branches that lead from the root node of the tree to the interior node, together with missing-value tests for the data fields mentioned in the tests associated with the tree branches that emanate from the interior node.

Claim 3 recites "wherein, when an additional data field is incorporated into the construction of a subordinate model, an alternate subordinate model is constructed for use when said additional data field has a missing value.

Claim 4 defines "wherein a missing value is estimated by performing a prediction based on the known data values."

These (and the other dependent claims 5-20) exemplarily define elements and limitations which further place the invention squarely in the realm of statutory subject matter and which provide a useful, tangible and concrete result.

Thus, claims 2-20 are further statutory subject matter for a U.S. Letters Patent.

IX. CONCLUSION

In view of the foregoing, Appellants submit that claims 1-20, all the claims presently pending in the application, are directed to statutory subject matter and are clearly and patentably distinct form the prior art of record and in condition for allowance. Thus, the Board is respectfully requested to remove the rejections of claims 1-20.

Please charge any deficiencies and/or credit any overpayments necessary to enter this paper to Assignee's Deposit Account number 50-0510.

Respectfully submitted,

Dated: 11/06/03

Sean M. McGinn
Reg. No. 34,386

McGinn & Gibb, P.C.
8231 Old Courthouse Road, Suite 200
Vienna, VA 22882-3817
(703) 761-4100
Customer Number: 21254

APPENDIX

1. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for constructing a predictive model that can be used to make predictions even when the values of some or all inputs are missing or are otherwise unknown, the method comprising:

(5) presenting a collection of training data comprising examples of input values that are available to the model together with corresponding desired output value(s) that the model is intended to predict;

(6) generating a plurality of subordinate models, that together comprise an overall model, in such a way that:

each subordinate model has an associated set of application conditions that must be satisfied in order to apply the subordinate model when making predictions, the application conditions comprising:

i) tests for missing values for all, some, or none of the inputs,

and

ii) tests on the values of all, some, or none of the

inputs that are applicable when the values of the inputs mentioned in the tests have known values;

and

for at least one subordinate model, the training cases used in the construction of that subordinate model include some cases that indirectly satisfy the application conditions such that the application conditions are satisfied only after replacing one or more known data values in these training cases with missing values; and

outputting a specification of at least one of said subordinate models thus generated and making a prediction based on said at least one of said subordinate models thus-generated.

2. A device according to claim 1, wherein step (2) comprises generating a plurality of subordinate models such that the plurality cannot be arranged into a decision-tree hierarchy in such a way that:

(1) each branch of the tree corresponds to a test on the values of one or more data fields that can be satisfied only when those data fields have known values;

(2) each leaf of the tree corresponds to a subordinate model whose application conditions are defined by the conjunction of the tests along the branches that lead from the root node of the tree to the leaf node;

(3) the root node of the tree corresponds to a subordinate model whose application conditions include missing-value tests for the data fields mentioned in the tests associated with the tree branches that emanate from the root node;

and

(4) each interior node of the tree other than the root node corresponds to a subordinate model whose application conditions are defined by the conjunction of the tests along the branches that lead from the root node of the tree to the interior node, together with missing-value tests for the data fields mentioned in the tests associated with the tree branches that emanate from the interior node.

3. The program storage device according to claim 1, wherein, when an additional data field is incorporated into the construction of a subordinate model, an alternate subordinate model is constructed for use when said additional data field has a missing value.

4. The program storage device according to claim 1, wherein a missing value is estimated by performing a prediction based on the known data values.

5. The program storage device according to claim 1, wherein each subordinate model has an application condition that must be satisfied for said each subordinate model to be applied, and

wherein said application condition includes at least one of the values to be input to the model being missing.

6. The program storage device according to claim 1, wherein said outputting comprises outputting a specification of a plurality of subordinate models and their associated application conditions, and reading said specification being readable by the machine.

7. The program storage device according to claim 1, wherein said values are missing at random.

8. The program storage device according to claim 1, wherein based on said data collection, it is determined whether missing data values are missing at random or whether missing values convey information.

9. The program storage device according to claim 1, wherein a determination of randomness of missing values is made by examining the data values present.

10. The program storage device according to claim 1, wherein statistical tests are employed to determine randomness of missing values.

11. The program storage device according to claim 1, wherein randomness of missing values is assessed with a cross-validation technique.

12. The program storage device according to claim 11, wherein applying the cross validation technique comprises:

selecting and holding aside portions of the training cases that directly satisfy the application conditions of a subordinate model for validation purposes;

constructing first and second models using remaining training cases that directly satisfy the application conditions but were not held aside, such that one of the first and second models is constructed based only on the remaining cases and the second model is constructed based on the remaining cases plus the training cases that indirectly satisfy the application conditions;

estimating prediction errors of the first and second models by applying the models to the training cases held aside for validation purposes;

if a predictive accuracy of the first model is greater than that of the second model with a predetermined sufficiently high statistical significance, then assuming that missing values in the relevant fields are informative and the subordinate model should be constructed only from those training cases that directly satisfy the application conditions of the subordinate model; and

if a predictive accuracy of the first model is greater than that of the second model with a predetermined sufficiently high statistical significance, then missing values are treated as random events and the training cases that directly or indirectly satisfy the application conditions are used in the construction of the subordinate model.

13. The program storage device according to claim 12, wherein the cross-validation method further comprises:

if a subordinate model is constructed for use when two or more data fields have missing values, then missing values of some of these data fields are treated as missing at random and others of said data fields are treated as informative,

wherein the training cases constructing the subordinate model includes those that directly satisfy the application conditions of the subordinate model together with those that indirectly satisfy the application conditions when known data values are replaced with missing values, but only for those data fields for which missing values are to be treated as missing at random.

14. The program storage device according to claim 13, wherein determining whether said missing values should be treated as missing at random or which should be treated as informative, includes:

constructing a model assuming that all missing values are to be treated as informative, such that the model is constructed from those training cases that directly satisfy the application conditions of the subordinate model but are not being held aside for validation purposes, said model being termed the "current model";

for each missing value in the "current model" that is treated as informative, constructing another model that treats that missing value as missing at random while treating all other missing values in the same manner as the "current model";

of the new models, choosing the one model that yields the greatest predictive accuracy on the training cases defined in said constructing that were used to construct the first "current model," and calling this new model the "current model";

repeating the constructing of the another model and the choosing until all missing values are treated as missing at random by the "current model";

of all "current models" obtained in the constructing of the "current model" and choosing, choosing the model that yields the greatest predictive accuracy on the training cases held aside for validation purposes, and calling this model the "best model"; and

constructing the subordinate model, without holding training cases aside for validation purposes, using the same treatments of missing values used in the construction of the "best model."

15. The program storage device according to claim 1, wherein a determination as to how to treat missing values for subordinate models is deferred.

16. The program storage device according to claim 1, wherein if a top-down method is employed to construct the subordinate models, then the plurality of models include a single subordinate model that does not use any data fields as input and which has an application condition that is always true.

17. The program storage device according to claim 1, wherein if a bottom-up method is employed to construct the subordinate model, then the plurality of models include a plurality of subordinate models and application conditions, the application conditions covering all possible combinations of values of the data fields.

18. The program storage device according to claim 1, wherein training cases are ignored only if they contain missing values in data fields that are required not to have missing values by the application conditions of the subordinate model being constructed.

19. The program storage device according to claim 1, wherein data fields that contain missing values are ignored in the construction of only subordinate models, and

wherein a missing value deemed to be informative is treated as a legitimate data value

20. The program storage device according to claim 1, wherein said method is devoid of filing in missing values with an imputation procedure, a weighting scheme to compensate for the presence of missing data, and introduction of free parameters into the subordinate model to represent missing data.

data file: BOSTON1.STA [506 cases with 14 variables]
Data analyzed by Lim, Loh, and Shih (1997)

FILE INFO PAGE:
Data originally reported in Harrison & Rubinfeld (1978). PRICE is recoded into LOW, MEDIUM, and HIGH classifications. NOXLEVEL is recoded into low, medium, high. Variables have been assigned more intuitive names: PRICE = MEDV, ON_RIVER = CHAS, CRIME_RT = CRIM, %BIGLOTS = ZN, %INDUSTY = INDUS, NOXLEVEL = NOX, AVGNUMRM = RM, %OLDBLDG = AGE, DIST2WRK = DIS, HWYACCES = RAD, TAX_RATE = TAX, CLASSIZE = PTRATIO, RACESKEW = B, %LOWINCM = LSTAT.

VARIABLE SPECIFICATIONS:

No	Name	Format	MD Code	Long Label
1	PRICE	8.0	-9999	Median value of owner-occupied homes (Hi, Med, Low)
2	ON_RIVER	6.0	-9999	Indicator (1 if tract bounds Charles River; else 0)
3	CRIME_RT	9.5	-9999	Per capita crime rate by town
4	%BIGLOTS	6.1	-9999	Percent residential land zoned for lots > 25k sq.ft.
5	%INDUSTY	7.2	-9999	Percent of non-retail business acres per town
6	NOXLEVEL	7.0	-9999	Concentration of nitric oxides (high, medium, low)
7	AVGNUMRM	8.3	-9999	Average number of rooms per dwelling
8	%OLDBLDG	6.1	-9999	Percent of owner-occupied units built prior to 1940
9	DIST2WRK	7.3	-9999	Weighted distances to five Boston employment centers
10	HWYACCES	5.1	-9999	Index of accessibility to radial highways
11	TAX_RATE	7.1	-9999	Full-valued property tax rate per $10,000
12	CLASSIZE	7.2	-9999	Pupil-teacher ratio by town
13	RACESKEW	8.2	-9999	1000*(Bk-0.63)^2 where Bk is the fraction of blacks
14	%LOWINCM	8.3	-9999	Percent lower status of the population

Exhibit 1

data file: BOSTON1.STA [506 cases with 14 variables]
Data analyzed by Lim, Loh, and Shih (1997)

VARIABLES AND THEIR TEXT VALUES:

Var 1: PRICE - Median value of owner-occupied homes (Hi, Med, Low) (-9999)

Text	Numeric	Long label
LOW	1	
MEDIUM	2	
HIGH	3	

Var 6: NOXLEVEL - Concentration of nitric oxides (high, medium, low) (-9999)

Text	Numeric	Long label
low	1	
medium	2	
high	3	

Exhibit 2

data file: BOSTON1.STA [506 cases with 14 variables]
Data analyzed by Lim, Loh, and Shih (1997)

	1 PRICE	2 ON_RIV	3 CRIME_RT	4 %BIGLO	5 %INDUST	6 NOXLEVE	7 AVGNUMRM	8 %OLDBL	9 DIST2WR	10 HWYAC	11 TAX_RAT	12 CLASSIZ	13 RACESKEW	14 %LOWINCM
1	HIGH	0	.00632	18.0	2.31	medium	6.575	65.2	4.090	1.0	296.0	15.30	396.90	4.980
2	MEDIUM	0	.02731	0.0	7.07	low	6.421	78.9	4.967	2.0	242.0	17.80	396.90	9.140
3	HIGH	0	.02729	0.0	7.07	low	7.185	61.1	4.967	2.0	242.0	17.80	392.83	4.030
4	HIGH	0	.03237	0.0	2.18	low	6.998	45.8	6.062	3.0	222.0	18.70	394.63	2.940
5	HIGH	0	.06905	0.0	2.18	low	7.147	54.2	6.062	3.0	222.0	18.70	396.90	5.330
6	HIGH	0	.02985	0.0	2.18	low	6.430	58.7	6.062	3.0	222.0	18.70	394.12	5.210
7	MEDIUM	0	.08829	12.5	7.87	medium	6.012	66.6	5.561	5.0	311.0	15.20	395.60	12.430
8	HIGH	0	.14455	12.5	7.87	medium	6.172	96.1	5.951	5.0	311.0	15.20	396.90	19.150
9	LOW	0	.21124	12.5	7.87	medium	5.631	100.0	6.082	5.0	311.0	15.20	386.63	29.930
10	MEDIUM	0	.17004	12.5	7.87	medium	6.004	85.9	6.592	5.0	311.0	15.20	386.71	17.100
11	LOW	0	.22489	12.5	7.87	medium	6.377	94.3	6.347	5.0	311.0	15.20	392.52	20.450
12	MEDIUM	0	.11747	12.5	7.87	medium	6.009	82.9	6.227	5.0	311.0	15.20	396.90	13.270
13	MEDIUM	0	.09378	12.5	7.87	medium	5.889	39.0	5.451	5.0	311.0	15.20	390.50	15.710
14	MEDIUM	0	.62976	0.0	8.14	medium	5.949	61.8	4.708	4.0	307.0	21.00	396.90	8.260
15	LOW	0	.63796	0.0	8.14	medium	6.096	84.5	4.462	4.0	307.0	21.00	380.02	10.260
16	MEDIUM	0	.62739	0.0	8.14	medium	5.834	56.5	4.499	4.0	307.0	21.00	395.62	8.470
17	MEDIUM	0	1.05393	0.0	8.14	medium	5.935	29.3	4.499	4.0	307.0	21.00	386.85	6.580
18	LOW	0	.78420	0.0	8.14	medium	5.990	81.7	4.258	4.0	307.0	21.00	386.75	14.670
19	MEDIUM	0	.80271	0.0	8.14	medium	5.456	36.6	3.797	4.0	307.0	21.00	288.99	11.690
20	LOW	0	.72580	0.0	8.14	medium	5.727	69.5	3.797	4.0	307.0	21.00	390.95	11.280
21	LOW	0	1.25179	0.0	8.14	medium	5.570	98.1	3.798	4.0	307.0	21.00	376.57	21.020
22	MEDIUM	0	.85204	0.0	8.14	medium	5.965	89.2	4.012	4.0	307.0	21.00	392.53	13.830
23	LOW	0	1.23247	0.0	8.14	medium	6.142	91.7	3.977	4.0	307.0	21.00	396.90	18.720
24	LOW	0	.98843	0.0	8.14	medium	5.813	100.0	4.095	4.0	307.0	21.00	394.54	19.880
25	LOW	0	.75026	0.0	8.14	medium	5.924	94.1	4.400	4.0	307.0	21.00	394.33	16.300
26	LOW	0	.84054	0.0	8.14	medium	5.599	85.7	4.455	4.0	307.0	21.00	303.42	16.510
27	LOW	0	.67191	0.0	8.14	medium	5.813	90.3	4.682	4.0	307.0	21.00	376.88	14.810
28	LOW	0	.95577	0.0	8.14	medium	6.047	88.8	4.453	4.0	307.0	21.00	306.38	17.280
29	LOW	0	.77299	0.0	8.14	medium	6.495	94.4	4.455	4.0	307.0	21.00	387.94	12.800
30	MEDIUM	0	1.00245	0.0	8.14	medium	6.674	87.3	4.239	4.0	307.0	21.00	380.23	11.980
31	LOW	0	1.13081	0.0	8.14	medium	5.713	94.1	4.233	4.0	307.0	21.00	360.17	22.600
32	LOW	0	1.35472	0.0	8.14	medium	6.072	100.0	4.175	4.0	307.0	21.00	376.73	13.040
33	LOW	0	1.38799	0.0	8.14	medium	5.950	82.0	3.990	4.0	307.0	21.00	232.60	27.710
34	LOW	0	1.15172	0.0	8.14	medium	5.701	95.0	3.787	4.0	307.0	21.00	358.77	18.350
35	LOW	0	1.61282	0.0	8.14	medium	6.096	96.9	3.760	4.0	307.0	21.00	248.31	20.340

Exhibit 3

Classification Tree for PRICE

Number of splits = 4; Number of terminal nodes = 5

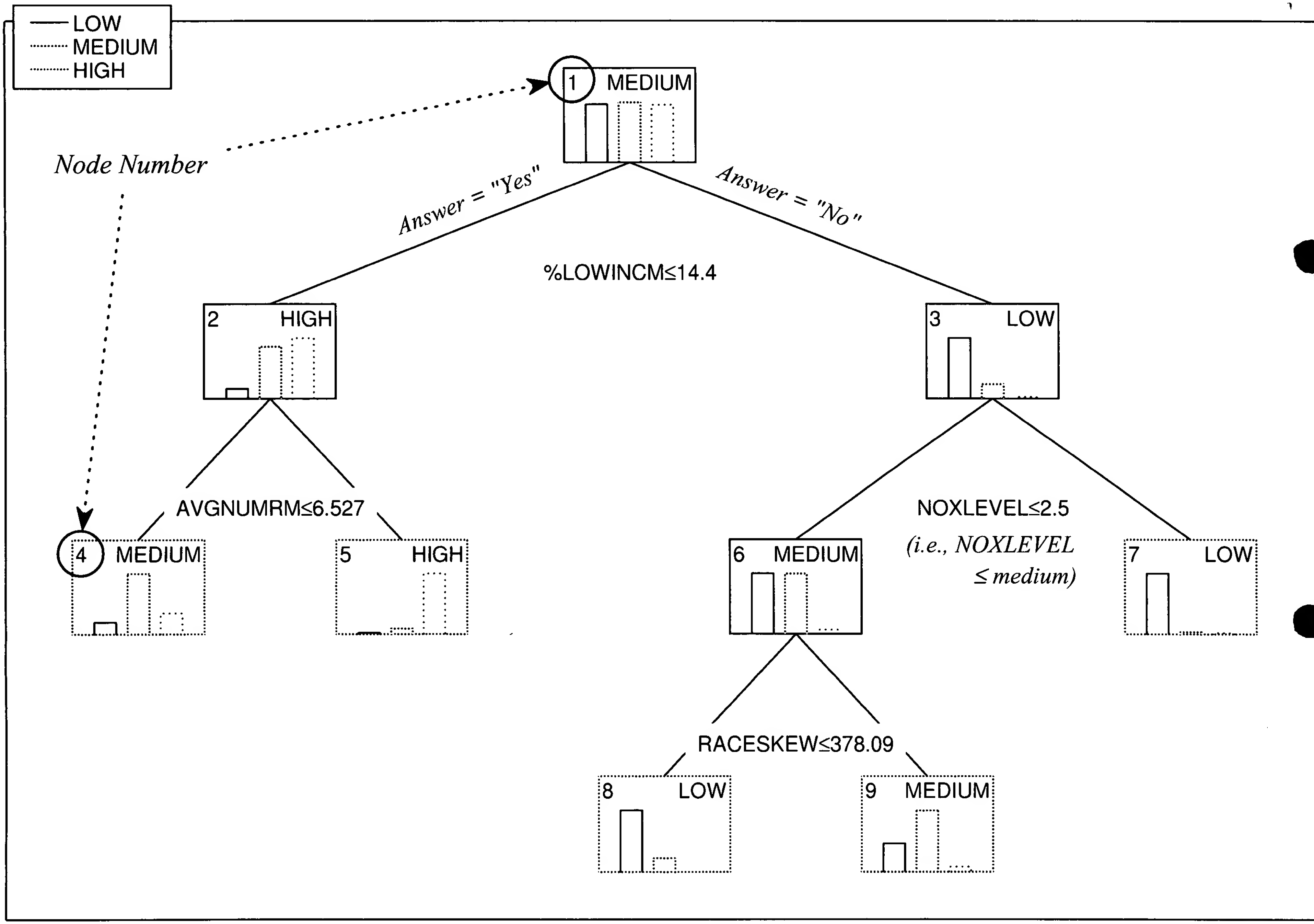

Exhibit 4

Classification Tree for PRICE

Number of splits = 6; Number of terminal nodes = 7

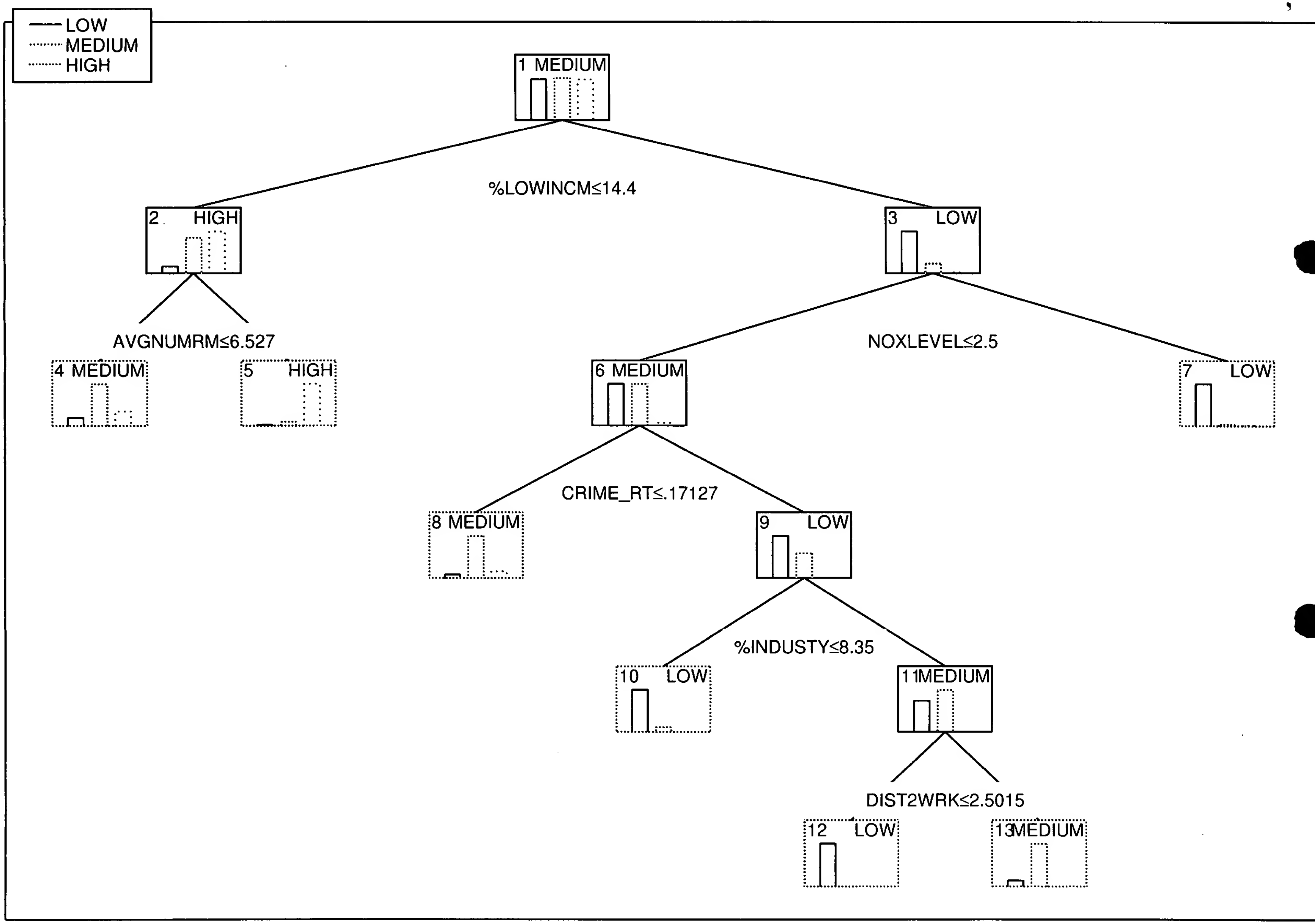

Exhibit 5

Classification Tree for NOXLEVEL

Number of splits = 7; Number of terminal nodes = 8

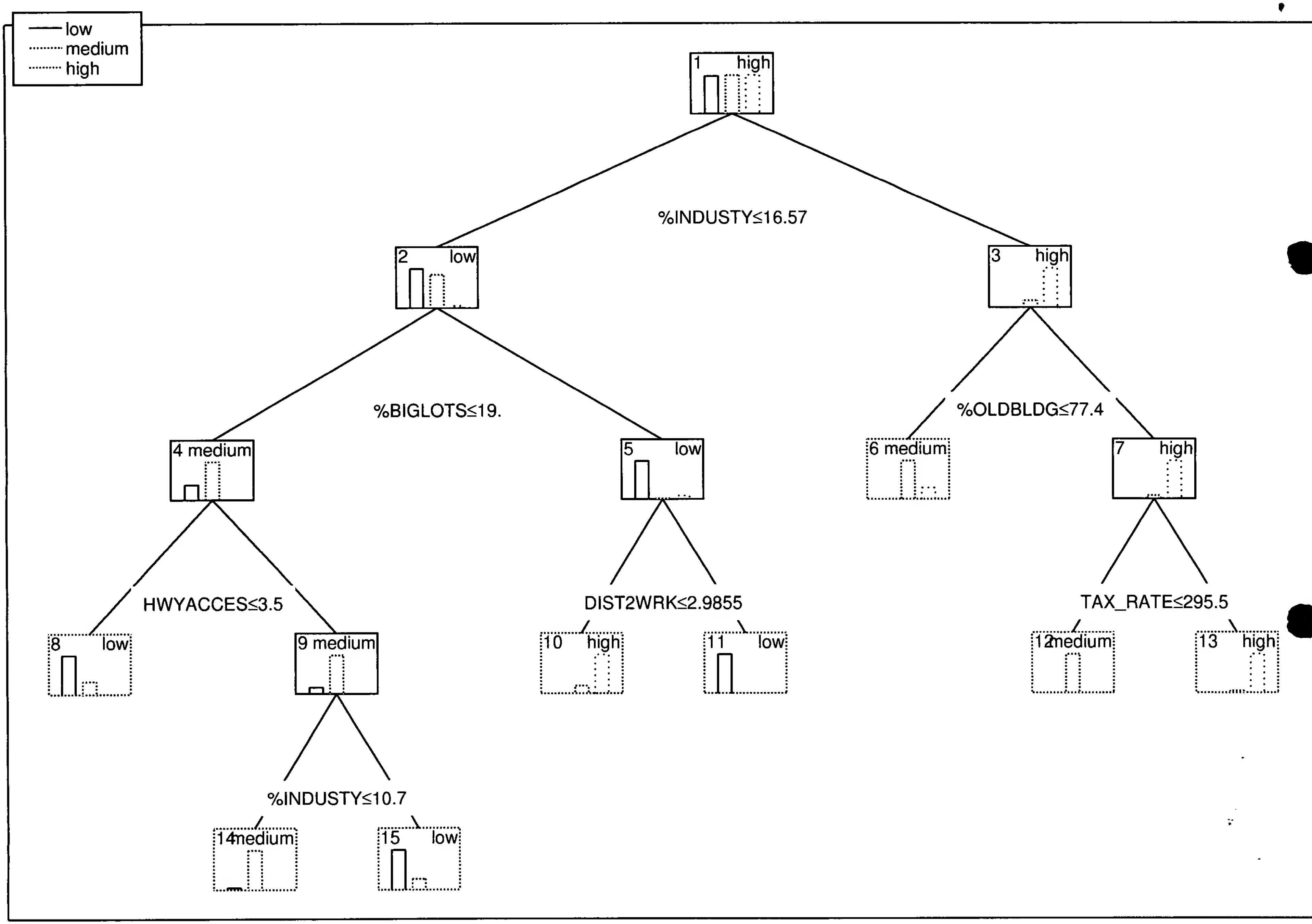

Exhibit 6